

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-210400
 (43)Date of publication of application : 11.08.1995

(51)Int.Cl.

G06F 9/46

(21)Application number : 06-019894
 (22)Date of filing : 20.01.1994

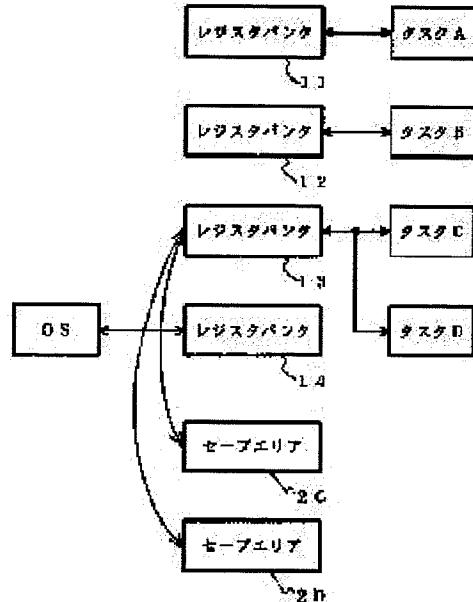
(71)Applicant : SONY CORP
 (72)Inventor : FUJISHIRO TOSHIHIKO
 OIWA JUNJI

(54) PROCESSOR FOR MULTI-TASK PROCESSING

(57)Abstract:

PURPOSE: To shorten dispatch time by saving and recovering the context of respective tasks in corresponding save areas by changeover of register banks and protecting it.

CONSTITUTION: The register banks 11 and 12 provided in a central processing unit are fixedly allocated to the tasks A and B in one-to-one correspondence and the register bank 13 is allocated to the tasks C and D in common. When a processing is shifted to the tasks C and D, the contents are saved in the corresponding save areas 2C and 2D and the saved contents are recovered from the save area 2D or 2C corresponding to the task to the remaining register bank 13. Thus, the tasks C and D using the register bank 13 are operated similarly to the tasks A and B allocated to the register banks 11 and 12 in one-to-one correspondence. Thus, the dispatch time in a real time operating system is shortened and the preparation of applications is not limited.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-210400

(43)公開日 平成7年(1995)8月11日

(51)Int.Cl.⁶
G 0 6 F 9/46

識別記号 庁内整理番号
3 1 3 D 7629-5B

F I

技術表示箇所

審査請求 未請求 請求項の数3 FD (全6頁)

(21)出願番号

特願平6-19894

(22)出願日

平成6年(1994)1月20日

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 藤代 俊彦

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

(72)発明者 大岩 純二

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

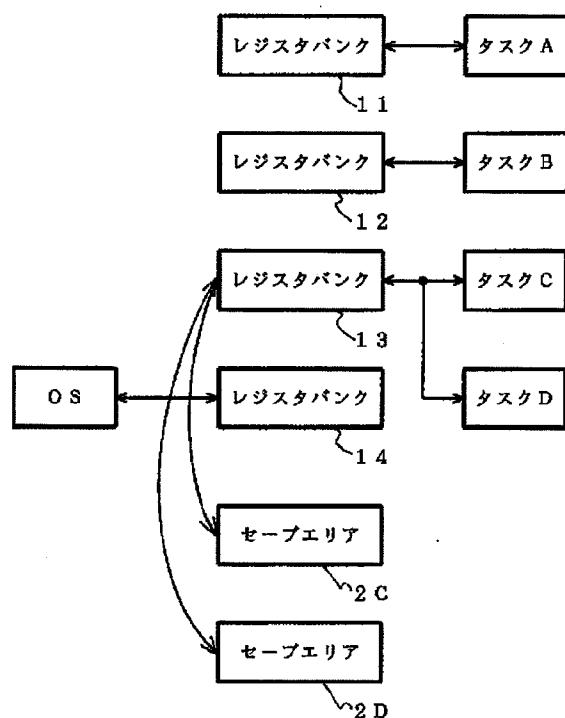
(74)代理人 弁理士 佐藤 正美

(54)【発明の名称】 マルチタスク処理の処理装置

(57)【要約】

【目的】 リアルタイムオペレーティングシステムのコ
ンピュータにおいて、ディスパッチ時間を短縮する。

【構成】 中央処理装置に4つのレジスタバンク11～
14を設ける。レジスタバンク11、12を、タスク
A、Bに1対1で固定的に割り当てる。レジスタバンク
13を、タスクC、Dに共通に割り当てる。レジスタバ
ンク13を使用するタスクCあるいはDに処理が移行す
るとき、レジスタバンク13の内容を、この内容のタス
クCあるいはDに対応するセーブエリア2Cあるいは2
Dに退避する。レジスタバンク13を使用するタスクに
対応するセーブエリア2Dあるいは2Cから、ここに退
避してある内容を残りのレジスタバンク13に復帰す
る。



【特許請求の範囲】

【請求項1】 リアルタイムオペレーティングシステムのコンピュータにおいて、中央処理装置がp個($p \geq m+1$ 。 $m \geq 1$)のレジスタバンクを有し、このp個のレジスタバンクのうちのm個のレジスタバンクを、m個のタスクに1対1で固定的に割り当て、上記p個のレジスタバンクのうちの残りのレジスタバンクを、n個($n \geq 2$)のタスクに共通に割り当て、上記残りのレジスタバンクを使用するタスクに処理が移行するとき、この残りのレジスタバンクの内容を、この内容のタスクに対応するセーブエリアに退避するとともに、

上記残りのレジスタバンクを使用するタスクに対応するセーブエリアから、ここに退避してある内容を上記残りのレジスタバンクに復帰するようにしたマルチタスク処理の処理装置。

【請求項2】 請求項1に記載のマルチタスク処理の処理装置において、

上記残りのレジスタバンクを使用するタスクに処理が移行するとき、上記残りのレジスタバンクを前回使用したタスクと、上記残りのレジスタバンクを今回使用するタスクとが異なるときのみ、上記内容の退避及び復帰を実行するようにしたマルチタスク処理の処理装置。

【請求項3】 請求項1あるいは請求項2に記載のマルチタスク処理の処理装置において、

$p = m+1$ となるようにしたマルチタスク処理の処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、リアルタイムオペレーティングシステムにおけるマルチタスク処理の処理装置に関する。

【0002】

【従来の技術】コンピュータのプログラムは、通常、シーケンシャルに実行されていくので、例えば2つの処理を同時に実行するためには、それらを所定の管理プログラムにより並行に処理する必要がある。

【0003】この並行に処理する部分をタスクと呼び、2つ以上のタスクを並行に時分割処理する技術を、マルチタスク処理と呼んでいる。そして、制御用のコンピュータのオペレーティングシステムは、一般に、このマルチタスク処理の機能を有している。

【0004】また、組み込み機器に使用されるオペレーティングシステムで重要な項目として、リアルタイム性があげられるが、このリアルタイム性を考慮して作成されたオペレーティングシステムは、リアルタイムオペレーティングシステムと呼んでいる。そして、このリアルタイムオペレーティングシステムの提供する機能は、以下の項目に分類することができる。

① タスク管理機能

タスクの状態を直接操作したり、参照したりする機能。

② タスク付属同期機能

タスクをサスペンド状態にして一時的に実行を停止する機能、それを解除する機能など、タスクにあらかじめ附加されている同期機能。

③ 同期・通信機能

セマフォ、イベントフラグ、メイルボックスを用いた、タスクとは独立した同期機能及び通信機能。

④ メモリプール管理機能

カーネルによってメモリブロックの割り当て及びメモリプールの管理を行う機能。

⑤ 割り込み管理機能

外部割り込みの禁止・許可及び外部割り込みに対するハンドラの定義を行う機能。

⑥ 時間管理機能

システムロックの設定や参照する機能、タスクを遅延させる機能及び指定した時刻に起動されるハンドラの定義を行う機能。

⑦ システム管理機能

カーネルのバージョンを参照する機能。

【0005】そして、リアルタイムオペレーティングシステムの提供する上記の機能を利用するため、アプリケーションからオペレーティングシステムに対して処理要求をすることを、システムコールを発行すると言っている。

【0006】さらに、オペレーティングシステムがタスクを管理するための制御領域を、TCB (Task Control Block) と呼んでいる。このTCBは、各タスクを管理するために、タスクごとに必要となる。そして、TCBには、以下の情報が必要となる。

① タスクID

タスクに割り当てられたID番号

② タスクステータス

タスクの状態

③ タスク優先度

タスク起動時のタスクの優先度

④ スタックポインタ

タスクに割り当てられたスタックポインタ

⑤ コンテキスト

レジスタの内容。プログラムステータスワードなど

⑥ 個別情報

また、マルチタスク処理を行うためには、タスクの状態として、少なくとも次の4つの状態が必要となる。

① 実行状態(RUN)

カーネルによって実行権が渡されたタスクの状態。

② 実行可能状態(READY)

カーネルから実行権が渡されれば、実行状態になれるタスクの状態。

③ 待ち状態(WAIT)

タスクが何らかの条件が満足されるのを待っている状態。実行条件が整えば実行可能状態となる。

④ 休止状態(DORMANT)

タスクが起動されていない状態、または終了後の状態。そして、アプリケーションのタスクから処理要求があつたときの処理の流れは、以下のとおりである。

1. タスクから処理要求を行う。
2. リアルタイムオペレーティングシステムの該当する処理要求の中でコンテキストなどの退避を行う。
3. 処理要求の処理を行う。
4. 実行可能状態のタスクから次に実行するタスクをサチする(スケジューリング)。
5. 実際に実行するためにコンテキストなどを復帰させる。
6. スケジューリングの結果のタスクに実行権を与える(ディスパッチ)。

【0007】なお、スケジューリングを行う場合、タスクが持っているタスク優先度を考慮してスケジューリングを行う必要がある。つまり、2つのタスクA、Bが実行可能状態で、先にタスクBが実行可能になったとしても、タスクAに高い優先度が与えられていたときには、タスクAに実行権が与えられる。以上が、リアルタイムオペレーティングシステムの概略である。

【0008】

【発明が解決しようとする課題】上述からも明らかに、リアルタイムオペレーティングシステムにおいては、システムコールに対する応答時間(アプリケーションがシステムコールを発行してから次のアプリケーションに処理が戻ってくるまでの時間。ディスパッチ時間)を短縮することが、重要である。すなわち、リアルタイムオペレーティングシステムにおいては、ディスパッチ時間を短縮することが、性能を評価する指標の1つとなる。

【0009】ところで、コンピュータの中央処理装置が動作するために必要な複数のレジスタを1つのグループとし、その複数のグループから構成されたものを、レジスタバンクと呼んでいる。

【0010】そして、上記2.項のコンテキストの退避は、そのコンテキストをレジスタバンクからそのタスクのセーブエリアに転送することにより実現され、上記5.項のコンテキストの復帰は、そのコンテキストをタスクのセーブエリアからレジスタバンクに転送することにより実現される。したがって、コンテキストの退避・復帰を行う時間は、コンテキストをレジスタバンクと各タスクのセーブエリアとの間で転送する時間となるが、この転送にはメモリのアクセスを伴うので、時間がかかり、ディスパッチ時間が長くなってしまう。

【0011】そこで、中央処理装置が複数のレジスタバンクを提供している場合、コンテキストをレジスタバンクとセーブエリアとの間で転送する代わりに、レジスタ

バンクを切り換えることによりコンテキストの退避・復帰を行うことが考えられている。すなわち、そのようにしても、あるタスクのコンテキストが次のタスクにより破壊されることはなく、かつ、コンテキストの保護に必要な時間を短縮できるので、結果として、ディスパッチ時間を短縮することができる。

【0012】しかし、中央処理装置が提供するレジスタバンクの数には制限があるので、1つのタスクに1つのレジスタバンクを割り当てる場合、並行して動作のできる

10 タスクの数は、レジスタバンクの数に制限されてしまい、空いているレジスタバンクがなければ、次のタスクを実行できないことになる。そして、このことは、アプリケーションを作成する上で制限事項となってしまう。

【0013】この発明は、このような点にかんがみ、リアルタイムオペレーティングシステムにおいて、ディスパッチ時間を短縮するとともに、特にアプリケーションの作成に制限を生じないようにしようとするものである。

【0014】

20 【課題を解決するための手段】今、並行して動作のできるタスクの数が、レジスタバンクの数により制限されず、タスクにレジスタバンクの空きを持つ状態がないようできれば、アプリケーション側に制限事項がなくなる。

【0015】そこで、この発明においては、複数のレジスタバンクの中から例えば1つのレジスタバンクを、複数のタスクに対して共通のレジスタバンクとして使用する。つまり、そのようにすれば、レジスタバンクの数がタスクの数より少なくとも、すべてのタスクを実行させることができる。

【0016】しかし、単に、レジスタバンクを共通に使用するだけでは、制御が次のタスクに移行したとき、それまで動作していたタスクのコンテキストが破壊されてしまう。また、共通のレジスタバンクを使用しているタスクが終了しないと、その共通のレジスタバンクを使用する他のタスクに、レジスタバンク待ちの状態が起きてしまい、ディスパッチ時間を短縮できない。

【0017】そこで、さらに、この発明においては、制御が共通のレジスタバンクを使用するタスクに移行する40とき、その共通のレジスタバンクの内容をタスクに固有のセーブエリアに退避させ、共通のレジスタバンクを開放する。また、移行先のタスクに固有のセーブエリアから、ここに退避してある内容を読み出して共通のレジスタバンクに復帰させる。

【0018】すなわち、この発明においては、各部の参考符号などを、後述する図1の実施例に対応させると、リアルタイムオペレーティングシステムのコンピュータにおいて、中央処理装置が4つのレジスタバンク11～14を有し、この4つのレジスタバンク11～14のうちの2つのレジスタバンク11、12を、2つのタスク

A、B に 1 対 1 で固定的に割り当てる、4 つのレジスタバンク 1 1 ～ 1 4 のうちの残りのレジスタバンク 1 3 を、2 つのタスク C、D に共通に割り当てる、残りのレジスタバンク 1 3 を使用するタスク C あるいはタスク D に処理が移行するとき、この残りのレジスタバンク 1 3 の内容を、この内容のタスク C あるいはタスク D に対応するセーブエリア 2 C あるいはセーブエリア 2 D に退避するとともに、残りのレジスタバンク 1 3 を使用するタスクに対応するセーブエリア 2 D あるいはセーブエリア 2 C から、ここに退避してある内容を残りのレジスタバンク 1 3 に復帰するようにしたものである。

【0019】

【作用】共通のレジスタバンク 1 3 を使用するタスク C、D が、レジスタバンク 1 1、1 2 が 1 対 1 で割り当てられているタスク A、B と同じように動作する。

【0020】

【実施例】図 1 に示す例においては、タスクが 4 つの場合であり、A、B、C、D が、その第 1 ～ 第 4 のタスクを示す。また、この例においては、これらタスク A ～ D の優先順位は、上記のタスク順であり、タスク A が最も高い優先順位であるとする。

【0021】さらに、この例においては、中央処理装置の提供するレジスタバンクも 4 つの場合であり、1 1、1 2、1 3、1 4 が、その第 1 ～ 第 4 のレジスタバンクを示す。

【0022】また、OS はリアルタイムオペレーティングシステムを示し、この例においては、4 つのレジスタバンク 1 1 ～ 1 4 のうち、レジスタバンク 1 4 が、オペレーティングシステム OS に固定的に割り当てる。

【0023】したがって、残る 3 つのレジスタバンク 1 1 ～ 1 3 が、4 つのタスク A ～ D に割り当てられることになるが、この例においては、タスク A、B は、この順に優先順位が高いので、1 1 がタスク A に固定的に割り当たられ、レジスタバンク 1 2 がタスク B に固定的に割り当たられる。そして、レジスタバンク 1 3 が、タスク C と、タスク D とに共通に割り当たられる。

【0024】さらに、1 つのレジスタバンク 1 3 が、2 つのタスク C、D に共通に割り当てられているので、タスク C あるいはタスク D が実行されるとき、レジスタバンク 1 3 のコンテキストを保護するための退避エリアとして、タスク C、D に固有のセーブエリア 2 C、2 D が設けられる。

【0025】そして、アプリケーションのタスクから処理要求があったときの処理の流れは、上記 1. ～ 6. 項のとおりであるが、この場合、制御が、タスク A ～ D の間で、どのような組み合わせで移行するかにしたがって、レジスタバンク 1 1 ～ 1 3 が、次のように切り換えられる。

A 制御が、あるタスクからタスク A あるいはタスク B へ移行する場合

この場合には、レジスタバンクが、それまで使用していたレジスタバンクからレジスタバンク 1 1 あるいはレジスタバンク 1 2 に切り換えられる。

【0026】そして、このとき、レジスタバンク 1 1、1 2 は、タスク A、B に 1 対 1 で固定的に割り当たされているので、上記のようなレジスタバンクの切り換えを行ってもタスク A、B の各コンテキストは保護される。

【0027】したがって、制御は、あるタスクからタスク A あるいはタスク B へ移行するとともに、このとき、タスク A あるいはタスク B への移行は、高速に実行されることになる。

B 制御が、タスク A あるいはタスク B からタスク C に移行する場合

この場合には、さらに、次の B-1 項及び B-2 項の場合がある。

B-1. タスク A あるいはタスク B の前にタスク C が実行されていた場合

この場合には、レジスタバンクが、それまで使用していたレジスタバンク 1 1 あるいはレジスタバンク 1 2 からレジスタバンク 1 3 に切り換えられる。

【0028】そして、レジスタバンク 1 3 は、タスク C、D に共通に割り当てられているが、今の場合、レジスタバンク 1 3 が、最後に使用されたとき、タスク C により使用されている。したがって、レジスタバンク 1 3 には、タスク C のコンテキストが残っている。

【0029】したがって、レジスタバンクを、それまで使用していたレジスタバンク 1 1 あるいはレジスタバンク 1 2 からレジスタバンク 1 3 に切り換えることにより、タスク C に移行することができる。また、タスク A、B、D の各コンテキストも保護される。

B-2. タスク A あるいはタスク B の前にタスク D が実行されていた場合

レジスタバンク 1 3 は、タスク C、D に共通に割り当たされているとともに、今の場合、レジスタバンク 1 3 が、最後に使用されたとき、タスク D により使用されている。したがって、レジスタバンク 1 3 には、タスク D のコンテキストが残っている。

【0030】そこで、この場合には、レジスタバンク 1 3 のコンテキストが、タスク D のセーブエリア 2 D に転送されて退避され、次に、タスク C のセーブエリア 2 C から、ここに退避されているタスク C のコンテキストが読み出され、これがレジスタバンク 1 3 に転送されて復帰される。

【0031】そして、その後、レジスタバンクが、それまで使用していたレジスタバンク 1 1 あるいはレジスタバンク 1 2 からレジスタバンク 1 3 に切り換えられる。

【0032】したがって、制御はタスク C に移行することができる。さらに、タスク A、B、D の各コンテキストも保護される。

C 制御が、タスク A あるいはタスク B からタスク D に

移行する場合

この場合は、B. 項において、タスクC及びセーブエリア2 Cと、タスクD及びセーブエリア2 Dとが入れ換わるだけで、B. 項と同様である。すなわち、やはり、次のC-1項及びC-2項の場合がある。

C-1. タスクAあるいはタスクBの前にタスクDが実行されていた場合

この場合には、レジスタバンクが、それまで使用していたレジスタバンク1 1あるいはレジスタバンク1 2からレジスタバンク1 3に切り換えられる。

【0033】そして、今の場合、レジスタバンク1 3が、最後に使用されたとき、タスクDにより使用されているので、レジスタバンク1 3には、タスクDのコンテキストが残っている。

【0034】したがって、レジスタバンクを、それまで使用していたレジスタバンク1 1あるいはレジスタバンク1 2からレジスタバンク1 3に切り換えることにより、タスクDに移行することができる。また、タスクA、B、Cの各コンテキストも保護される。

C-2. タスクAあるいはタスクBの前にタスクCが実行されていた場合

レジスタバンク1 3は、タスクC、Dに共通に割り当てられているとともに、今の場合、レジスタバンク1 3が、最後に使用されたとき、タスクCにより使用されている。したがって、レジスタバンク1 3には、タスクCのコンテキストが残っている。

【0035】そこで、この場合には、レジスタバンク1 3のコンテキストが、タスクCのセーブエリア2 Cに転送されて退避され、次に、タスクDのセーブエリア2 Dから、ここに退避されているタスクDのコンテキストが読み出され、これがレジスタバンク1 3に転送されて復帰される。

【0036】そして、その後、レジスタバンクが、それまで使用していたレジスタバンク1 1あるいはレジスタバンク1 2からレジスタバンク1 3に切り換えられる。

【0037】したがって、制御はタスクDに移行することができる。さらに、タスクA、B、Cの各コンテキストも保護される。

D. 制御が、タスクCからタスクDに移行する場合

今の場合、レジスタバンク1 3が、最後に使用されたとき、タスクCにより使用されているので、レジスタバンク1 3には、タスクCのコンテキストが残っている。

【0038】そこで、この場合には、レジスタバンク1 3のコンテキストが、タスクCのセーブエリア2 Cに転送されて退避され、次に、タスクDのセーブエリア2 Dから、ここに退避されているタスクDのコンテキストが読み出され、これがレジスタバンク1 3に転送されて復帰される。

【0039】したがって、制御はタスクDに移行することができる。さらに、タスクA、B、Cの各コンテキス

トも保護される。

E. 制御が、タスクDからタスクCに移行する場合

この場合は、D. 項において、タスクC及びセーブエリア2 Cと、タスクD及びセーブエリア2 Dとが入れ換わるだけで、D. 項と同様である。すなわち、今の場合、レジスタバンク1 3が、最後に使用されたとき、タスクDにより使用されているので、レジスタバンク1 3には、タスクDのコンテキストが残っている。

【0040】そこで、この場合には、レジスタバンク1 3のコンテキストが、タスクDのセーブエリア2 Dに転送されて退避され、次に、タスクCのセーブエリア2 Cから、ここに退避されているタスクCのコンテキストが読み出され、これがレジスタバンク1 3に転送されて復帰される。

【0041】したがって、制御はタスクCに移行することができる。さらに、タスクA、B、Dの各コンテキストも保護される。

【0042】図1の例においては、以上のようにしてリアルタイムオペレーティングシステムにおいて、マルチタスク処理が実現される。

【0043】図2に示す例においては、タスクが、タスクA～Eの5つであり、中央処理装置の提供するレジスタバンクが、レジスタバンク1 1～1 4の4つの場合である。なお、タスクA～Eの優先順位は、上記のタスク順であり、タスクAが最も高いものとする。

【0044】そして、この例においても、4つのレジスタバンク1 1～1 4のうち、レジスタバンク1 4が、オペレーティングシステムOSに固定的に割り当てられる。また、1 1がタスクAに固定的に割り当てられ、レジスタバンク1 2がタスクBに固定的に割り当てられる。そして、レジスタバンク1 3が、タスクCと、タスクDと、タスクEと共に共通に割り当てられる。

【0045】さらに、レジスタバンク1 3が、タスクC～Eに共通に割り当てられているので、タスクC、DあるいはタスクEが実行されるとき、レジスタバンク1 3のコンテキストを保護するための退避エリアとして、タスクC、D、Eに固有のセーブエリア2 C、2 D、2 Eが設けられる。

【0046】そして、アプリケーションのタスクから処理要求があったとき、図1の例と同様に制御が実行される。

【0047】すなわち、制御が、あるタスクからタスクAあるいはタスクBに移行する場合には、上記A項のように、レジスタバンク1 1あるいはレジスタバンク1 2に切り換えられる。

【0048】また、制御が、タスクAあるいはタスクBからタスクC～Eのどれかに移行する場合には、上記B-1項あるいはC-1項のように、レジスタバンク1 3のコンテキストが変化しないときには、レジスタバンク1 3に切り換えられる。

【0049】しかし、上記B-2項あるいはC-2項のように、レジスタバンク13のコンテキストが変化するときには、レジスタバンク13のコンテキストが、セーブエリア2C～2Eのうちの対応するセーブエリアに退避されるとともに、処理要求のあったタスクに対応するセーブエリアに退避してあるコンテキストがレジスタバンク13に復帰され、その後、レジスタバンクが、レジスタバンク13に切り換えられる。

【0050】さらに、制御が、タスクC～Eの間で移行する場合には、上記D項あるいはE項のように、レジスタバンク13のコンテキストが、セーブエリア2C～2Eのうちの対応するセーブエリアに退避されるとともに、処理要求のあったタスクに対応するセーブエリアに退避してあるコンテキストがレジスタバンク13に復帰される。

【0051】したがって、この例においては、4つのレジスタバンク11～14により5つのタスクA～Dを実行することができ、マルチタスク処理が実現される。

【0052】なお、上述においては、1つのレジスタバンクを2つ以上のタスクに共通に割り当てるが、i個($i \geq 2$)のレジスタバンクのそれぞれを2つ以上のタスクに共通に割り当てることができる。

【0053】

【発明の効果】この発明によれば、レジスタバンクの切*

*り換えにより各タスクのコンテキストを保護するようにしているので、デイスパッチ時間を短縮することができる。

【0054】しかも、その場合、特にこの発明によれば、m個($m \geq 1$)のレジスタバンクをm個のタスクに1対1で固定的に割り当てる、他のレジスタバンクをn個($n \geq 2$)のタスクに共通に割り当てるとともに、この共通のレジスタバンクを使用するタスクに処理が移行するとき、そのコンテキストを、対応するセーブエリアに

10 対して退避・復帰するようにしている。

【0055】したがって、レジスタバンクの数が制限されているても、並行して動作のできるタスクの数に制限を受けることがない。さらに、レジスタバンクが空いているかどうかを考慮する必要がなく、アプリケーションの作成に制限を与えることがない。

【図面の簡単な説明】

【図1】この発明の一例を示す図である。

【図2】この発明の他の例を示す図である。

【符号の説明】

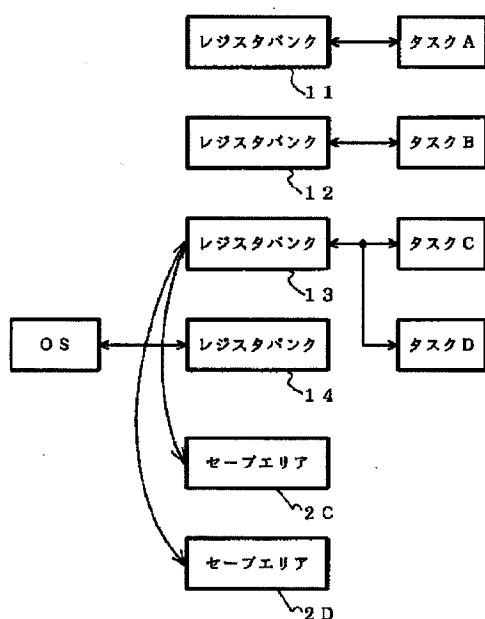
20 11～14 レジスタバンク

2C～2E セーブエリア

A～E タスク

OS リアルタイムオペレーティングシステム

【図1】



【図2】

